REMARKS

Applicants' representative would like to thank Examiner Kimberly Flynn for kindly taking a substantial amount of time on September 29, 2004 to discuss the merits of the subject invention. Applicants' representative is aware of the time constraint that is placed on the Examiner and is appreciative of the Examiner's willingness to devote such large quantity of time to discuss the case on the merit.

In view of the following discussion, the Applicants submit that none of the claims now pending in the application are obvious under the provisions of 35 U.S.C. § 103. The Applicants believe that all of these claims are now in allowable form.

I. REJECTIONS OF CLAIMS 1-6, AND 9-34 UNDER 35 U.S.C. § 103

A. Claims 1-4, 6-21, 25-29 and 34

The Examiner rejected claims 1-4, 6-21, 25-29 and 34 under 35 U.S.C. §103(a) as being unpatentable over Stevens (TCP/IP Illustrated Volume 1, Chapters 2, 18, 19, 20 and 21) in view of Belove et al. (U.S. Patent No. 5,491,820, issued February 13, 1996, hereinafter Belove) and further in view of Baylor et al. (6,175,899, issued January 16, 2001, hereinafter Baylor). The rejection is respectfully traversed.

Steven is a textbook that teaches the TCP/IP architecture and protocols.

Belove teaches a distributed objected-oriented database management system. Specifically, Belove teaches an object-oriented approach to storage and transmission of retrievable items in a client-server computer. (See Belove, Abstract)

Baylor teaches a method for providing virtual atomicity in a multi processor environment having access to multilevel caches. Specifically, Baylor teaches a caching scheme where an invalidation signal is received at a higher level cache location, which then sends invalidation signals to all lower level caches which store portions of the invalidated line. (See Baylor, Abstract)

The Examiner's attention is directed to the fact that Steven, Belove and Baylor (either singly or in any permissible combination) fails to disclose or suggest a method and apparatus for updating information in a low-bandwidth client/server object-oriented system, where the server determines the updates that are required by a client, while the client is simply listening for the updates, as positively claimed by the Applicants. Applicants' independent claims 1, 9, 14, 18, 21, 25 and 34 positively recite:

> 1. A method for transmitting a packet of data from a first computing system to a second computing system, the first computing system and the second computing system being included in a client/server object-based computing system, wherein the first computing system is a server and the second computing system is a client, the method comprising:
>
> identifying the packet of data using the first computing system, wherein said second computing system is listening, wherein the packet of data includes data which represents an object in the client/server object-based computing system, the object been identified as an object which the second computing system has an interest in receiving updates;
>
> attempting to send the packet of data from the first computing system to the second computing system;
>
> determining when the packet of data is received by the second computing system; and
>
> sending an acknowledgment from the second computing system to the first computing system when it is determined that the packet of data is received by the second computing system, the acknowledgement being arranged to indicate that the packet of data is received by the second computing system. (Emphasis added)
>
> 9. A method for transmitting a packet of data from a first computing system to a second computing system, the first computing system and the second computing system being included in a client/server object-based computing system, wherein the first computing system is a server and the second computing system is a client, the method comprising:
>
> a) attempting to send the packet of data from the first computing system to the second computing system, wherein said second computing system is listening, wherein the packet of data includes data which represents an object in the client/server object-based computing system, the object been identified as an object which the second computing system has an interest in receiving updates;
>
> b) determining when the packet of data is received by the second computing system;
>
> c) identifying the packet of data as being successfully sent when it is determined that the packet of data is received by the second computing system; and
>
> d) assuming that packet losses have occurred when it is determined that the packet of data is not received by the second computing system, wherein assuming that packet losses have occurred includes repeating a) and b) for up to a predetermined number of times. (Emphasis added)

14. A computer program product for transmitting a packet of data from a first computing system to a second computing system, the first computing system and the second computing system being included in a client/server object-based computing system, wherein the first computing system is a server and the second computing system is a client, the computer program product comprising:

computer code for identifying the packet of data using the first computing system, wherein said second computing system is listening, wherein the packet of data includes data which represents an object in the client/server object-based computing system, the object been identified as an object which the second computing system has an interest in receiving updates;

computer code for attempting to send the packet of data from the first computing system to the second computing system;

computer code for determining when the packet of data is received by the second computing system;

computer code for sending an acknowledgment from the second computing system to the first computing system when it is determined that the packet of data is received by the second computing system, the acknowledgement being arranged to indicate that the packet of data is received by the second computing system; and

a computer readable medium that stores the computer codes. (Emphasis added)

18. A computer program product for transmitting a packet of data from a first computing system to a second computing system, the first computing system and the second computing system being included in a client/server object-based computing system, wherein the first computing system is a server and the second computing system is a client, the computer program product comprising:

computer code for attempting to send the packet of data from the first computing system to the second computing system, wherein said second computing system is listening, wherein the packet of data includes data which represents an object in the client/server object-based computing system, the object been identified as an object which the second computing system has an interest in receiving updates;

computer code for determining when the packet of data is received by the second computing system;

computer code for identifying the packet of data as being successfully sent when it is determined that the packet of data is received by the second computing system;

computer code for assuming that packet losses have occurred when it is determined that the packet of data is not received by the second computing system, wherein assuming that packet losses have occurred includes computer code for re-attempting to send the packet of data from the first computing system to the second computing system and computer code for determining when the re-attempt to send the packet of data is successful for up to a predetermined number of times; and a computer readable medium that stores the computer codes. (Emphasis added)

21. A client/server object-based computing system, the client/server object-based computing system comprising:

at least one server;

at least one client, wherein the at least one client is listening and being at least periodically in communication with the server across a low-bandwidth communications channel;

a mechanism arranged to reduce statistical information associated with the client/server object-based computing system, the mechanism including a measuring system for measuring time elapsed for a packet of data to be sent by the at least one server to the at least one client, wherein the packet of data determined by the at least one server includes data which represents an object in the client/server object-based computing system, the object been identified as an object which the at least one client has an interest in receiving updates;

a data transmission system, the data transmission system being arranged to transmit data between the at least one client and the at least one server, the data transmission system further being arranged to repeatedly attempt to transmit the data for up to a number of times determined by the mechanism; and

a reconnection system, the reconnection system being arranged to attempt to reinstate the low-bandwidth communications channel after the transmission system repeatedly attempts to transmit the data for up to the number of times determined by the mechanism. (Emphasis added)

25. A method for substantially optimizing the transmission of data between a first computing system to a second computing system, the first computing system and the second computing system being included in a client/server object-based computing system, wherein the first computing system is a server and the second computing system is a client, the data including a first packet, the method comprising:

a) gathering statistical information associated with the client/server object-based computing system, wherein gathering the statistical information includes measuring time used to send at least a second packet of data between the first computing system and the second computing system, wherein the second packet of data includes data which represents an object in the client/server object-based computing system;

b) attempting to send the first packet from the first computing system to the second computing system, wherein said second computing system is listening, wherein the first packet of data includes data which represents a first object in the client/server object-based computing system, the first object been identified as an object which the second computing system has an interest in receiving updates;

c) determining when the first packet is received by the second computing system;

> d) determining an amount of time to elapse before attempting to re-send the first packet when it is determined that the first packet is not received by the second computing system, the amount of time being determined using the measured time used to send the at least second packet; and
>
> e) attempting to re-send the first packet after the amount of time elapses. (Emphasis added)
>
> 34. A method for transmitting a packet of data from a first computing system to a second computing system, the first computing system and the second computing system being included in a client/server object-based computing system, wherein the first computing system is a server and the second computing system is a client, the method comprising:
>
> identifying the packet of data using the first computing system, wherein said second computing system is listening, wherein the packet of data includes data which represents an object in the client/server object-based computing system, the object being represented in an object list in the first computing system, the object list arranged to include objects that are to be updated, and the object also being represented in a filter tree which is arranged to identify objects that the second computing system has an interest in;
>
> attempting to send the packet of data from the first computing system to the second computing system;
>
> determining when the packet of data is received by the second computing system; and
>
> sending an acknowledgment from the second computing system to the first computing system when it is determined that the packet of data is received by the second computing system, the acknowledgement being arranged to indicate that the packet of data is received by the second computing system. (Emphasis added)

In one embodiment, Applicants' invention teaches a method and apparatus where updated information on a client/server object-oriented computing system, is determined by the server and then forwarded to the client, where the client is simply listening for the updated information. In other words, the clients are not initiating update requests. Since communications between a server and a plurality of clients often consume a substantial portion of a communication channel, e.g., a low-bandwidth channel, it is important to minimize the amount of such communications. In Applicants' invention, if updated information is necessary to be forwarded to the clients, then it is the task of the server to determine the necessary updates and then to forward that updated information to the clients at the discretion and leisure of the server. The clients simply "listen" for such updated information. In other words, the clients need not inquire about any updated information from the server. This approach significantly conserves the available bandwidth of the communication channel, especially in a low-bandwidth channel. (See Applicants' specification, page 29, lines 15-24)

In contrast, neither Steven nor Belove teaches or suggests this novel approach. Often clients are required to initiate the request for updated information. (See Belove, Column 6, lines 7-36). Thus, if the system has a large number of clients, the server would be bombarded with update requests that will substantially consume the available bandwidth in a communication channel, especially in a low-bandwidth channel.

The Examiner attempted to bridge the gap left by Steven and Belove by combining Baylor to the cited references. The Examiner alleged that the cache locations in Baylor teach monitoring or snooping. Applicants respectfully disagree.

During the Examiner Interview of September 29, 2004, Applicants' representative presented several arguments. First, cache locations are <u>not</u> computing systems. Therefore, the constraints and criticalities addressed in a memory system are completely different than the present invention where there are computing systems.

Second, the Examiner is citing three references from very different fields, where Steven teaches the TCP/IP architecture and protocols, Belove teaches a distributed objected-oriented database management system, and Baylor teaches a caching scheme. Namely, the Examiner is combining teachings from the fields of transmission protocol, database management and memory management to make obvious Applicants' invention. It is submitted that such use of references from such diverse fields is an impermissible application of hindsight by the Examiner. For example, Baylor discloses a <u>bus based system</u> where the various caches are able to receive invalidation signals. However, Steven teaches TCP/IP architecture and protocols that operate on a <u>Wide Area Network (WAN)</u>. Applicants submit that communication criticalities addressed in a memory system is completely different than communication criticalities addressed in a WAN. It is submitted that such combination is impermissible.

In rejecting claims under 35 U.S.C. §103, it is incumbent upon the Examiner to establish a factual basis to support the legal conclusion of obviousness. <u>See</u> <u>In re Fine</u>, 837 F.2d 1071, 1073, 5 USPQ2d 1596, 1598 (Fed. Cir. 1988). In so doing, the Examiner is expected to make the factual determinations set forth in <u>Graham v. John Deere Co.</u>, 383 U.S. 1, 17, 148 USPQ 459, 467 (1966), and to provide a reason why one having ordinary skill in the pertinent art would have been led to modify the prior art or to combine prior art references to arrive at the claimed invention. Such reason must stem from some teaching, suggestion or implication in the prior art as a whole or knowledge generally available to one having ordinary skill in the art. Uniroyal, Inc. v. Rudkin-Wiley Corp., 837 F.2d 1044, 1051, 5 USPQ2d 1434, 1438 (Fed. Cir.), cert. denied, 488 U.S. 825 (1988); Ashland Oil, Inc. v. Delta Resins & Refractories, Inc., 776 F.2d 281 293, 227 USPQ 657, 664 (Fed. Cir. 1985), cert. Denied, 475 U.S. 1017 (1986); ACS Hosp. Sys., Inc. v. Montefiore Hosp. 732 F.2d 1572, 1577, 221 USPQ 929, 933 (Fed. Cir. 1984). These showings by the Examiner are an essential part of complying with the burden of presenting a prima facie case of obviousness. Note In re Oetiker, 977 F.2d 1443, 1445, 24 USPQ2d 1443, 1444 (Fed. Cir. 1992).

Applicants submit that the Examiner has failed to present a prima facie case of obviousness. Thus, independent claims 1, 9, 14, 18, 21, 25 and 34 are not made obvious by the teaching of Steven in view of Belove and Baylor.

Dependent claims 2-4, 6, 10-13, 15-17, 19-20, 22-24, and 26-29 depend, either directly or indirectly, from claims 1, 9, 14, 18, 21, and 25 and recite additional features thereof. As such and for the exact same reasons set forth above, the Applicants submit that claims 2-4, 6, 10-13, 15-17, 19-20, 22-24, and 26-29 are also not made obvious by the teaching of Steven in view of Belove and Baylor. Therefore, the Applicants submit that all these dependent claims also fully satisfy the requirements of 35 U.S.C. § 103 and are patentable thereunder.

B. Claims 5, 22-24, 30 and 31

The Examiner rejected claims 5, 22-24, 30 and 31 under 35 U.S.C. §103(a) as being unpatentable over Stevens (TCP/IP Illustrated Volume 1, Chapters 2, 18, 19, 20 and 21) and Belove et al. (U.S. Patent No. 5,491,820, issued February 13, 1996, hereinafter Belove) and further in view of Rich et al. (U.S. Patent No. 6,457,065, issued September 24, 2002, hereinafter Rich). The rejection is respectfully traversed.

Rich teaches a method, system, and computer program product for improving the performance of distributed object systems. Specifically, Rich teaches a remote object that is replicated to the node of the distributed system from which it is accessed. (See Rich, Abstract)

Again, the gap left by Steven in view of Belove is not bridged by Rich. Namely, Rich also does not teach the concept where the server determines the updates that are required by a client, while the client is simply listening for the updates. Thus, for the same reason as discussed above, independent claims 1, 21, and 25 are not made obvious by the teaching of Steven and Belove in view of Rich.

Dependent claims 5, 22-24, 30 and 31 depend, either directly or indirectly, from claims 1, 21, and 25 and recite additional features thereof. As such and for the exact same reasons set forth above, the Applicants submit that claims 5, 22-24, 30 and 31 are also not made obvious by the teaching of Steven and Belove in view of Rich. Therefore, the Applicants submit that all these dependent claims also fully satisfy the requirements of 35 U.S.C. § 103 and are patentable thereunder.

C. Claims 32 and 33

The Examiner rejected claims 32 and 33 under 35 U.S.C. §103(a) as being unpatentable over Stevens (TCP/IP Illustrated Volume 1, Chapters 2, 18, 19, 20 and 21) and Belove et al. (U.S. Patent No. 5,491,820, issued February 13, 1996, hereinafter Belove) and further in view of Mangold et al. (U.S. Patent No. 5,926,232, issued July 20, 1999, hereinafter Mangold). The rejection is respectfully traversed.

Mangold teaches a method for optimizing the transmission of signals, especially video signals, over a channel with a predetermined channel data rate after source encoding for data reduction and channel encoding, in which redundancy is added to the source-encoded signals for error control protection, at least one quality parameter of the transmitted and decoded signals is measured. Specifically, as a function of the measured quality parameter, the relationship between the source accuracy of the source encoding and the added redundancy is changed in opposite directions to optimize the transmission. (See Mangold, Abstract)

Again, the gap left by Steven in view of Belove is not bridged by Mangold. Namely, Mangold also does not teach the concept where the server determines the updates that are required by a client, while the client is simply listening for the updates.

Thus, for the same reason as discussed above, independent claim 25 is not made obvious by the teaching of Steven and Belove in view of Mangold.

Dependent claims 32 and 33 depend, either directly or indirectly, from claim 25 and recite additional features thereof. As such and for the exact same reasons set forth above, the Applicants submit that claims 32 and 33 are also not made obvious by the teaching of Steven and Belove in view of Mangold. Therefore, the Applicants submit that all these dependent claims also fully satisfy the requirements of 35 U.S.C. § 103 and are patentable thereunder.

Conclusion

Thus, the Applicants submit that none of the claims, presently in the application, is obvious under the provisions of 35 U.S.C. § 103. Consequently, the Applicants believe that all these claims are presently in condition for allowance. Accordingly, both reconsideration of this application and its swift passage to issue are earnestly solicited.

If, however, the Examiner believes that there are any unresolved issues requiring adverse final action in any of the claims now pending in the application, it is requested that the Examiner telephone Mr. Kin-Wah Tong at (732) 530-9404 so that appropriate arrangements can be made for resolving such issues as expeditiously as possible.

Respectfully submitted,

9/30/04

Kin-Wah Tong
Reg. No. 39,400
(732) 530-9404

Moser, Patterson & Sheridan, LLP
595 Shrewsbury Avenue
Suite 100
Shrewsbury, NJ 07702